Updated June 23, 1999

Implementing USB Wakeup & ACPI S3 on ICH-based Systems

Jerzy Kolinski – Intel Architecture Lab / Instantly Available PC System Architect

Ike Dean – Platform Driver Division Software Architect

Abstract

This document describes the recommended system requirements needed to enable USB wakeup from ACPI S3 state (Suspend to RAM). It specifies the components that need to be implemented on the motherboard in order to enable S3 wakeup. This document also describes the need for the power budgeting capability that is needed to ensure safe and reliable system operation.

It is assumed that the reader is familiar with ACPI architecture and the ICH controller HUB specifications.

Introduction

The previous generation of computer systems based on the PIIX4 I/O bridge were not capable of waking up from ACPI S3 state. It was not possible to design a system supporting ACPI S3 (suspend to RAM) state and still be able to wake up from USB devices such as keyboard, mouse or the modem.

With the introduction of the next generation of I/O controller 82801AA (ICH) and 82801AB (ICH0) components USB S3 wakeup is now possible. With the new architecture the wakeup detection logic of the USB controller is part of the resume well of the I/O controller and is powered from the auxiliary power source to detect USB S3 wakeup events.

However it is also required to properly design the motherboard to ensure USB wakeup. The USB devices must have valid 5V power while suspended in S3 state. The design of the power delivery system on the motherboard is described in the next section.

It is also required to ensure that there is enough standby power delivered by the power supply to ensure reliable system operation. This can be accomplished by ether specifying power supply with extended 5V standby capacity or by using power budgeting capability. More details are provided in the following sections.

Standby Capacity of ATX Power Supply vs USB Requirements

ATX Standby Power Capacity	USB Bus Power Requirements (Per USB Port on Motherboard with USB Device Attached)					
	Suspend		Waiting for Enumeration		Fully Active	
	Min	Max	Min	Max	Min	Max
720mA	200µA	2.5mA	1mA	100mA	1mA	500mA

A proposed addition to the USB Common Class Specification's Interface Power Management specification (IPM) has been made to decrease current needed during wakeup to 100 mA for remote wakeup devices that support IPM. According to the proposal, the remote wakeup device is not allowed to consume 500mA until host software commands it to return to its fully active state. However, use of IPM is optional for USB devices.

Windows 98 Defaults for Wake-up Devices

Several drivers that ship with Windows 98 support wake-up devices natively. Table 1 shows the defaults for which devices support wake-up and when the wake-up feature is enabled.

Table 1. Windows 98 Defaults for Wake-up Devices

Device	Desktop	Portable*
8042 (PS/2 style) keyboard**	Always enabled	Never enabled
8042 (PS/2 style) mouse**	Always enabled	Never enabled
ACPI buttons (lid, power, sleep)	Always enabled	Always enabled
COM ports/modems	When COM port open	When COM port open
RTC	On demand from applications	On demand from applications
USB root hub (device insertion/removal only)	Always enabled	Never enabled
USB keyboard	Always enabled	Never enabled
USB mouse	Always enabled	Never enabled
Other devices	Based on third-party driver	Based on third-party driver

* A portable is any system with a PC Card controller.
** Wake-up using 8042 keyboard and mouse is only supported using ACPI mechanisms: the device must have appropriate _PRW and _PSW objects, must set a GPE on wake, and WAK_STS must be cleared accordingly.

Current Win98 Wakeup Policy for USB is to ALWAYS ENABLE wakeup regardless of Sx state. USB Wakeups can be globally enabled/disabled via a registry flag.

Platform Design Options

Case 1 : USB Ports lose power during S3

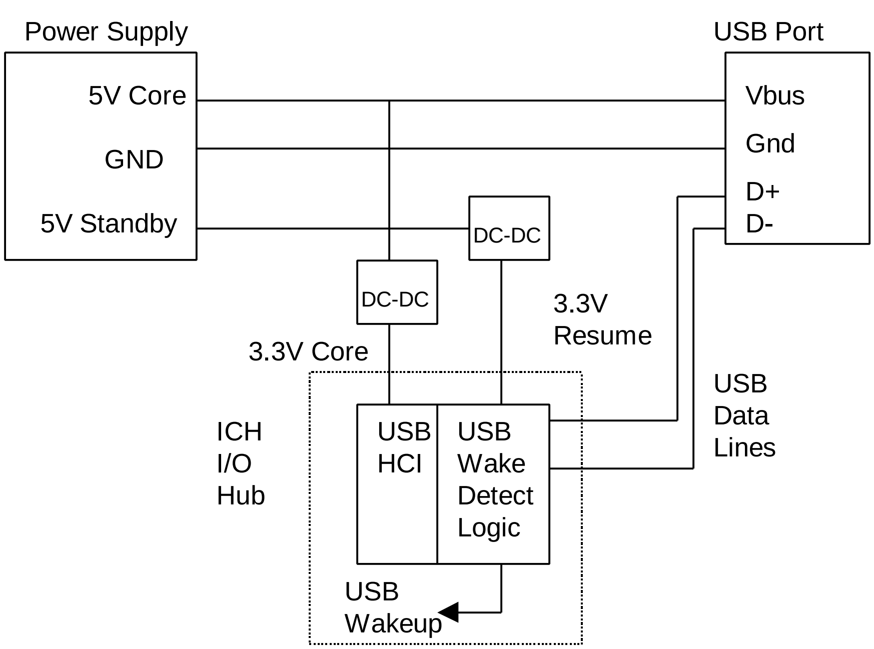
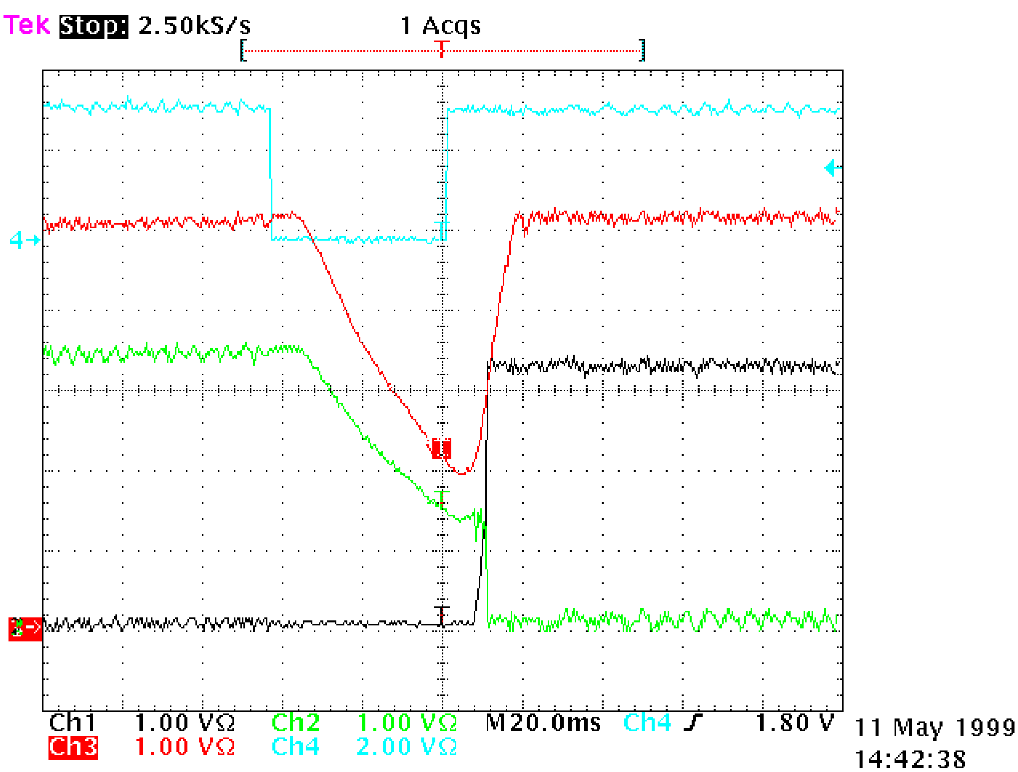


Figure 1. Typical Implementation of USB Power Subsystem

Due to a combination of the OS's default USB Wakeup Policy (ALWAYS ENABLED) and the platform's design of the USB Power Subsystem, this design prevents the platform from entering S3 when Bus Powered USB devices are connected to the USB Port.

While a bus powered usb device is plugged into the USB port, the system enters S3 and immediately resumes from S3, which effectively disables the use of S3 (ie STR) on the platform. As shown in the scope traces below, this behavior is due to the ICH detecting a valid disconnect event as power is being removed from the usb port. The behavior of the USB D+/D- signals is identical to what would be observed if when a bus-powered USB device is disconnected from the USB port.

This issue does not occur if there are Self-Powered USB devices connected to the USB port OR if USB Ports remain powered at the time the system is placed into S3.



Color	Channel	Signal	Description
Blue	4	SLP_S3#	Entry into S3 on falling edge, exit on rising edge. Exit due to ICH detecting valid disconnect event on USB D-/D+ signals
Red	3	USB V_{Bus}	Power to USB Port
Green	2	USB D-	USB Data
Black	1	USB D+	USB Data

Recommended BIOS Workaround for Case 1 to avoid resume from S3 issue with Bus Powered USB Peripherals:

One possible solution to this is to have the OS change it's USB Wakeup policy based on the type of USB peripheral plugged in (self powered or bus powered) AND based on how the platform's USB port is wired (standby or main power supply). However, since there is no mechanism for the OS to 'discovery' how the platform has it's USB port's 'wired', this is not practical. Therefore, the following additions to the platform's acpi bios are recommended for this case:

System BIOS implement an setup option that allows the user to enable/disable USB Wakeups from the S3 sleep state. The default setting for this BIOS option should be to disable USB wakeup from S3 under the assumption that majority of USB devices are bus-powered. Of course, this will limit the platforms ability to wake from USB, however, it does allow the platform to utilize S3. The assumption here is that proper usage of S3 gets preference over USB wakeups.

Note1: Simply declaring the USB device can only support wakeup from S1 is not acceptable because this will effectively disable S3....ACPI OS's use a 'least common denominator' method of determining what the deepest sleep state the platform can go into and still accommodate device wakeup power requirements. Since the default OS policy is to enable USB wakeups, if the USB controller indicates it can only support wake up from S1, and all other devices indicate they can support wakeup from S3, then the OS will resort to using S1 (instead of S3) to accommodate the USB device.

Note2: Suggested ACPI BIOS Implementation:

A) Add CMOS Setup Option:

USB Wakeup from S3: Enable/Disable

B) Modify _PTS Control Method:

If [(SLP_TYP=S3) & (USB Wakeup from S3 = Disable)]
Then [PORT0EN=0, PORT1EN=0]

where PORT0EN and PORT1EN are in the ICH USB Resume Enable register at Offset 0xC4h

C) Modify _WAK Control Method to always enable USB Resume Enable bits to allow detection of disconnect/connect events during runtime (S0)

[PORT0EN=1, PORT1EN=1]

Note3: Suggested APM BIOS Implementation:

A) Add CMOS Setup Option:

USB Wakeup from STR: Enable/Disable

B) SMM Code that does entry to STR.....checks CMOS setting and sets USB Controller's PORTxENs accordingly

C) SMM Code that does exit from STR....enables PORTxEN to allow detection of disconnect/connect events during run time.

Case 2: USB Ports maintain power during S3

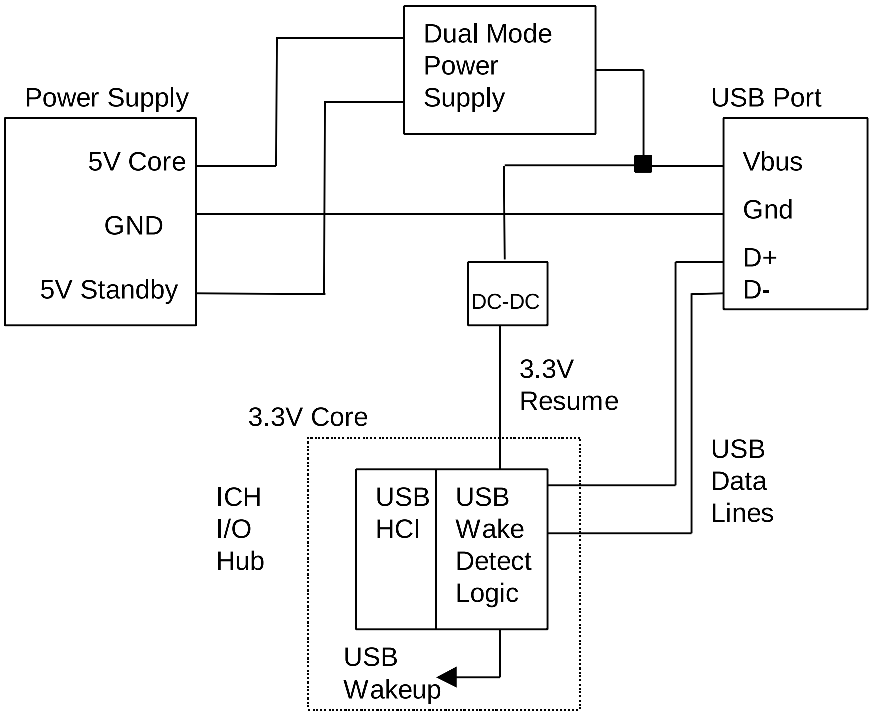


Figure 2: Recommended Implementation of USB Power Subsystem for Platforms that support S3

Based on the power measurements we have determined that majority of PCI and USB devices consume much less power than the maximum specified in USB and PCI PM specifications. Based on the actual power dissipation it is safe to use power supply rated at 720 mA with power budgeting capability. Power budgeting capability acts as the fuse to ensure that no more than 720 mA will be drawn from the power supply.

This implementation avoids the immediate resume from S3 issue when the system is used with bus powered USB peripherals by maintaining power to the USB connector.

In the implementation showed in Figure 2 the power at USB is still present in S3 state. The dual mode power delivery subsystem contains FET switches that provide the main power from VCC 5V rail during S0, S1 and S2 states and power from 5V standby rail in S3 (and S4) states.

The description of the dual mode power delivery circuit is provided at the Instantly Available PC home page at http://developer.intel.com/technology/iapc/downloads/fet_mb.htm.

System reliability issues

The remote wakeup device consumes 500 uA while in the suspend state and can consume up to 500 mA while in the working state. Currently if the USB device is enabled for wakeup it can transition from low power mode to high power mode instantaneously. On the other hand it can take up to 1 sec for the power supply to be fully operational and deliver required power.

Due to this fact the system designer must ensure that there is enough standby power capacity from the power supply for the transition period (~1s) to guarantee proper system operation.

For example for the system with 4 USB ports and no power budgeting capability one can calculate USB power needed as 4 x 500 mA = 2Amps.

Building power supplies with standby capacity greater than 2 A is expensive. This is the main reason that currently system designers disable USB wakeup from S3 state.

Power Budgeting capability

Power budgeting capability reads current PCI and USB configuration to determine the required power needed to safely enter and exit ACPI S3 state. If it determines that, there is not enough power available, for example the S1 state could be entered instead of S3 state.

Based on the power measurements the majority of PCI and USB devices consume much less power than the maximum specified in USB and PCI PM specifications. Based on the actual power dissipation it is safe to use power supply rated at 720 mA with power budgeting capability. Power budgeting capability acts as the fuse to ensure that no more than 720 mA will be drawn from the power supply.

It is planned that power budgeting capability will be part of the future releases of the operating system.

Conclusion

It is recommended to implement S3 USB wakeup for the systems with ICH I/O controller. To guarantee proper USB wakeup operation it is required to use power supply with min of 720 mA. It is also necessary to use power budgeting capability that acts as the system fuse.

In case when motherboard does not route Vdual voltage to USB connectors USB S3 wakeup should be disabled in the BIOS or the OS to prevent system from instantaneous wakeup due to the power transitions.